Harmonizing Textual and Graphical Visualizations of Domain Specific Models

Colin Atkinson and Ralph Gerbig
Agenda

- Motivation & Target
- Harmonized Graphical and Textual Editing
- Demo
Motivation

- **Domain-specific languages are widely spread**
  - **Textual:** HTML, CSS, BPEL, …
  - **Graphical:** Electric Circuits, BPMN, …
Motivation

- Domain-specific Language **Workbenches are highly focused** towards
  - **Text-based languages**: EMFText, Spoofax, JetBrains MPS, Xtext

- **Graphical Languages**: Generic Modeling Environment, Graphical Modeling Framework, Poseidon for DSLs, MetaEdit+
Sometimes one wants to edit in a mixed textual and graphical editor

If graphical and textual domain-specific workbench base on the same infrastructure, they can be wired together e.g. XText + GMF

Non-trivial glue code needs to be created to wire these two technologies up
A domain-specific language workbench which allows to create graphical and textual DSLs

- Using one tool and one language definition mechanism
- Allow simultaneous editing in textual and graphical editor without synchronization problems and visual glitches
- A workbench that makes changes to abstract and concrete syntax immediately available
- Domain-specific languages featuring more than one type/instance level
Agenda

- Motivation & Target
- Harmonized Graphical and Textual Editing
- Demo
Ingredients

- Multi-level Modeling
- Separation of Abstract and Concrete syntax definition
- Projectional textual and graphical editors

from http://www.kultiversum.de/
Multi-level Modeling

- Arbitrary number of classification levels
- Ontological and linguistic classification
  - Traits and Attributes
- Clabject
- Potency

![Diagram showing multi-level modeling concepts with labels and relationships between categories and levels.](image-url)
Separation of Abstract and Concrete Syntax

- Model elements are annotated with visualizers
- One element can have multiple visualizers attached
- Visualizers are available for textual and graphical languages
- Visualizer search algorithm finds visualizer to use for rendering
Projectional Editing
Agenda

■ Motivation & Target
■ Harmonized Graphical and Textual Editing
■ Demo
MelanEE - Screencast
Conclusions

- The harmonization of graphical and domain-specific languages can be achieved by wiring up existing technologies
  - Annotation of meta-models with concrete syntax visualization information (EMFText)
  - Rendering search algorithm across level boundaries (Meta-model Syntactic Sheets [2-level only])
  - Projectional Editors for Text (JetBrains MPS)
  - Meta-models to define graphical (GMF, MetaEdit+) and textual concrete syntaxes (EBNF, Antlr, EMFText, Spoofax)

- We are currently ...
  - refining the graphical and textual editors
  - experimenting with the import of EBNFs and try to extend our editor towards the expressiveness of EBNF
Thank You!